

CLAIMS

1 1. A method for controlling the synchrony of interprocess
2 communication (IPC) which uses: a. the original sender of
3 the communication, called the source; b. the ultimate
4 receiver of the communication, called the destination; c.
5 a kernel process that is capable of delivering
6 communications and unblocking the source of the
7 communication; and d. one or more processes, called
8 monitors, to which the kernel process may choose to
9 deliver a communication to, rather than the destination,
10 and which have the ability to request control of when a
11 source is unblocked, called the controlling monitor of
12 the source. wherein the method for controlling the
13 synchrony of interprocess communication comprise the
14 following steps:
15 d. when the communication is sent by the source, the
16 source is blocked by the kernel process;
17 e. the kernel either delivers the communication to the
18 destination or to a monitor;
19 f. the monitor may request that the kernel process make
20 it the controlling monitor of the source;
21 g. the monitor forwards the communication to the
22 destination specifying the source of the communication,
23 and the monitor is blocked by the kernel process until
24 the communication is delivered either to the destination
25 or another monitor;
26 h. when the communication is delivered to the
27 destination, the kernel process either: (1) signals the
28 controlling monitor of the specified source, if any, or
29 (2) unblocks the specified source;

30 i. the controlling monitor of the source can request that
31 the kernel process unblock the source at any time after
32 being signalled that the communication was delivered to
33 the destination.

1 2. A method as claimed in claim 1, wherein the monitor
2 may change the identity of the source that is seen by the
3 destination while maintaining the identity of the
4 original source for unblocking purposes.

1 3. A method as claimed in claim 1, wherein the monitor
2 may change the identity of the destination to which the
3 communication is to be delivered.

001100 THE 9900
1 4. A method as claimed in claim 1, wherein the message
2 embodied in the communication is not delivered to the
3 monitor.

1 5. A method as claimed in claim 1, wherein the timeouts
2 on the communication are interpreted as being relative to
3 the behaviors of the source and destination.

001100 THE 9900
1 6. A method as claimed in claim 5, wherein the kernel
2 process does not deliver the communication to the
3 destination or a monitor until the destination is ready
4 to receive the communication and the destination timeout
5 (i.e., the timeout set by the source on the amount of
6 time it will wait for the destination to become ready to
7 receive the communication) has not expired.

1 7. A method as claimed in claim 5, wherein the kernel
2 process delivers the communication to the monitor as soon

3 as the monitor is ready, and the monitor checks for the
4 destination timeout's expiration.

1 8. A method as claimed in claim 5, wherein the kernel
2 process verifies the source timeout (i.e., the timeout
3 set by the destination on the amount of time it will wait
4 for the source to initiate the communication) has not
5 expired before sending the communication to the
6 destination or a monitor.

1 9. A method as claimed in claim 1, wherein multiple
2 monitor processes may claim to be the controlling monitor
3 of a source.

1 10. A method as claimed in claim 1, wherein the kernel
2 process may authorize a monitor's permission to be the
3 controlling monitor of a particular source.

1 11. A method as claimed in claim 10, wherein the kernel
2 process may authorize a monitor's permission to be the
3 controlling monitor of any source.

1 12. A method as claimed in claim 1, wherein the
2 controlling monitor for a particular source may be stored
3 in the kernel.

1 13. A method as claimed in claim 9, wherein a sequence of
2 controlling monitors for a particular source may be
3 stored in the kernel.

1 14. A method as claimed in claim 1, wherein the identity
2 of the controlling monitor of a particular source is
3 passed in the IPC to the destination.

1 15. A method as claimed in claim 14, wherein the sequence
2 of controlling monitors for a particular source is stored
3 by the controlling monitor storing its controlling
4 monitor predecessor for each source.

1 16. A method as claimed in claim 1, wherein the
2 controlling monitor for a source is implemented by
3 changing the original source to the controlling monitor
4 and the controlling monitor stores the identity of the
5 original source.

1 17. A method as claimed in claim 16, wherein a sequence
2 of controlling monitors for a particular source is
3 implemented as a sequence of original source changes in
4 the monitors where the last is the true original source.

1 18. A method as claimed in claim 1, wherein the monitors
2 are implemented as threads in the same process.

1 19. A method as claimed in claim 1, wherein the monitors
2 are implemented as procedures in the same process.

1 20. A method as claimed in claim 1, wherein the monitor
2 procedures are in the kernel process.

1 21. A method of performing interprocess communications
2 (IPCs), comprising the steps of:

3 an IPC process receiving IPC requests, each of
4 the IPC requests including a source identifier
5 identifying a source and a destination identifier
6 identifying a destination;
7 building IPCs in response to the requests;
8 transmitting the IPCs from the sources to the
9 destinations;
10 intercepting and examining selected ones of the
11 IPCs; and
12 controlling the synchrony of the IPCs so that
13 each IPC appear to the source and to the destination to
14 be implemented according to the same semantics regardless
15 of whether the IPC is intercepted and examined.

1 22. A method according to Claim 21, wherein:

2 the step of building and transmitting IPCs
3 includes the step of using a kernel to build and transmit
4 the IPCs;

5 the step of intercepting and examining selected
6 ones of the IPCs includes the step of using monitors to
7 intercept and examine the selected ones of the IPCs; and

8 the controlling step includes the step of using
9 the monitors as extensions of the kernel so that the IPCs
10 appear to the sources and to the destinations to be
11 implemented according to the same semantics regardless of
12 whether a monitor is used or not used to intercept and
13 examine the IPCs.

1 23. A method according to claim 22, wherein, for each
2 IPC, one or more of the monitors manages the identity of
3 the source for the IPC.

4 the means for intercepting and examining
5 selected ones of the IPCs include monitors to intercept
6 and examine the selected ones of the IPCs; and
7 the monitors are used as extensions of the
8 kernel so that the IPCs appear to the sources and to the
9 destinations to be implemented according to the same
10 semantics regardless of whether a monitor is used or not
11 used to intercept and examine the IPCs.

1 28. A system according to claim 26, wherein, for each
2 IPC, one or more of the monitors manages the identity of
3 the source for the IPC.

1 29. A system according to Claim 26, wherein, for each
2 IPC, the source for the IPC is blocked at selected times,
3 and one or more of the monitors manages when the source
4 is unblocked.

1 30. A system according to Claim 26, wherein, for each
2 IPC, one of the monitors manages whether the monitor is
3 notified when the IPC reaches the destination for the
4 IPC.

1 31. A program storage device readable by machine,
2 tangibly embodying a program of instructions executable
3 by the machine to perform method steps for performing
4 interprocess communications (IPCs), said method steps
5 comprising:

6 an IPC process receiving IPC requests, each of
7 the IPC requests including a source identifier
8 identifying a source and a destination identifier
9 identifying a destination;

10 building IPCs in response to the requests;
11 transmitting the IPCs from the sources to the
12 destinations;
13 intercepting and examining selected ones of the
14 IPCs; and
15 controlling the synchrony of the IPCs so that
16 each IPC appear to the source and to the destination to
17 be implemented according to the same semantics regardless
18 of whether the IPC is intercepted and examined.

1 32. A program storage device according to Claim 31,
2 wherein:

3 the step of building and transmitting IPCs
4 includes the step of using a kernel to build and transmit
5 the IPCs;

6 the step of intercepting and examining selected
7 ones of the IPCs includes the step of using monitors to
8 intercept and examine the selected ones of the IPCs; and

9 the controlling step includes the step of using
10 the monitors as extensions of the kernel so that the IPCs
11 appear to the sources and to the destinations to be
12 implemented according to the same semantics regardless of
13 whether a monitor is used or not used to intercept and
14 examine the IPCs.

1 33. A program storage device according to claim 32,
2 wherein, for each IPC, one or more of the monitors
3 manages the identity of the source for the IPC.

1 34. A program storage device according to Claim 32,
2 further comprising the step of, for each IPC, blocking
3 the source for the IPC at selected times, and wherein one

4 or more of the monitors manages when the source is
5 unblocked.

1 35. A program storage device according to Claim 32,
2 wherein, for each IPC, one of the monitors manages
3 whether the monitor is notified when the IPC reaches the
4 destination for the IPC.

[illegible]